Checking the Convexity of Polytopes and the Planarity of Subdivisions

Olivier Devillers Giuseppe Liotta
Franco Preparata and Roberto Tamassia

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-97-11
December 1997

Checking the Convexity of Polytopes and the Planarity of Subdivisions*

Olivier Devillers[†] *Giuseppe Liotta*[‡]
Franco P. Preparata[§] *Roberto Tamassia*[§]

December 11, 1997

Abstract

This paper considers the problem of verifying the correctness of geometric structures. In particular, we design optimal checkers for convex polytopes in two and higher dimensions, and for various types of planar subdivisions, such as triangulations, Delaunay triangulations, and convex subdivisions. Our checkers are simpler and more general than the ones previously described in the literature. Their performance is analyzed also in terms of the algorithmic degree, which characterizes the arithmetic precision required.

*Research supported in part by the National Science Foundation under grant CCR–9423847, by the U.S. Army Research Office under grant DAAH04–96–1–0013, and by the EC ESPRIT Long Term Research Project ALCOM-IT under contract 20244 and CGAL under contract 21957.

[†]INRIA, BP 93, 06902 Sophia Antipolis Cedex, France. Work by this author performed in part while he was visiting the Center for Geometric Computing at Brown University. `Olivier.Devillers@sophia.inria.fr`

[‡]Dipartimento di Informatica e Sistemistica, Universitá di Roma "La Sapienza", Via Salaria 113, Roma 00198, Italy. Work by this author performed in part while he was visiting the Center for Geometric Computing at Brown University. `liotta@dis.uniroma1.it`

[§]Center for Geometric Computing, Department of Computer Science, Brown University, 115 Waterman Street, Providence, RI 02912–1910, USA. `{franco,rt}@cs.brown.edu`

1 Introduction

The development of checkers for geometric structures is justified by the expectation that it is easier to evaluate the quality of the output than the correctness of the algorithm producing it, and is further motivated by the increasing availability of geometric software on the Internet (see, e.g., [1]), and by the emerging client-server distributed models of geometric computing over the Web (see, e.g., [2]).

Mehlhorn *et al.* [19] identify three fundamental features of a good checker: *correctness*, *simplicity*, and *efficiency*.

In this paper, we consider checkers for subdivisions in two and higher dimensions. In particular, we consider two-dimensional planar subdivisions and convex polytopes in a fixed dimension d. The subdivision to be checked can be either a primitive structure, or a derived structure computed from a primitive one (e.g., the Voronoi diagram of a set of sites in the plane). More formally, we consider checkers whose input consists of:

- a geometric graph Γ (i.e., a graph with coordinates assigned to its vertices), which is claimed to induce a subdivision S;
- an optional primitive structure P, from which S is claimed to be derived;
- an optional *certificate* C, provided to facilitate the task of the checker; and
- a predicate $\mathcal{P}$ stating a property of S.

The task of the checker is to verify that a subdivision S is induced by geometric graph Γ and satisfies $\mathcal{P}$. The checker either *accepts* Γ, or *rejects* Γ producing evidence that S does not satisfy $\mathcal{P}$. Consider the following two examples:

- Γ is a graph with (x, y) coordinates assigned to the vertices; P is a set of points in the plane; C is a circular ordering of the edges incident on each vertex of Γ; and predicate $\mathcal{P}$ states that S is the Voronoi diagram of P.
- Γ is a graph with (x, y, z) coordinates assigned to the vertices; P and C are not defined; and predicate $\mathcal{P}$ states that S is a convex 3D polytope.

Clearly, the availability of a suitable certificate may be crucial to the efficiency of the checker. We consider three scenarios for the type of checker available:

Arbitrary-Certificate Scenario: This is the most favorable scenario for the checker. Namely, the checker can specify an arbitrary certificate C to be provided as additional input. The size $|C|$ of the certificate should be $O(|S|)$ if S is itself primitive, and $O(|S|+|P|)$ if S is derived from a primitive structure P. Also, if S is derived from P, the certificate should be computable as a byproduct of an optimal algorithm that constructs S from P without using additional asymptotic time or space. A checker that operates within the Arbitrary-Certificate Scenario is called an *A-checker*.

Consider, for example, the problem of verifying that a polygon Γ is the convex hull S of a set P of primitive points in the plane. A useful certificate C would consist, for each point p of P, either of a vertex of Γ coincident with p, or of a triplet a, b, c of vertices of Γ such that p is contained in the triangle $\triangle(a, b, c)$. It has been shown that there exists an A-checker using certificate C for verifying that a polygon Γ is the convex hull S of the points of a set P in linear time $O(|G|+|P|)$ (see, e.g., [21]).

Topology-Certificate Scenario: In this intermediate scenario, the checker has available a certificate C that describes the (claimed) topology of the subdivision S. A checker that operates within the Topology-Certificate Scenario is called a *T-checker*.

In the previous example of the planar convex-hull verification, a certificate C for the Topology-Certificate Scenario would consist simply of the circular sequence of the vertices of polygon Γ. With this certificate, it is possible to perform in linear time only a partial verification that polygon Γ is the convex hull S of the points of P. Namely, one can verify in time $O(|G|)$ that Γ is a convex polygon (see Section 3). However, verifying that the points of P that are not vertices of Γ are interior to Γ requires time $\Omega(\log |G|)$ per point.

For the more complex problem of verifying that a 3D geometric graph Γ realizes a convex polytope S, a certificate C for the Topology-Certificate Scenario would be a data structure that describes a planar embedding of Γ, e.g., circularly-sorted adjacency lists.

No-Certificate Scenario: This is the least favorable scenario for the checker. Namely, no certificate C is available to the checker, which must perform the verification using only geometric graph Γ. A checker that operates within the No-Certificate Scenario is called an *N-checker*.

The Arbitrary-Certificate Scenario follows the program checking paradigm pioneered by Blum and Kannan [4]. On the negative side, it requires that the algorithms constructing the subdivision be modified to produce the specified certificate. On the positive side, A-checkers are often faster and simpler to implement than other types of checkers, and their correctness is usually easily established. Sullivan *et al.* [21] show A-checkers for planar convex hull, sorting, and shortest path algorithms. A-checkers for d-dimensional convex hulls are also discussed in [19].

The Topology-Certificate Scenario requires a natural type of certificate, which many algorithms for constructing subdivisions are likely to produce by default. Mehlhorn *et al.* [19] present a T-checker for certifying the convexity of a d-dimensional polytope and mention T-checkers for several other geometric structures, including Delaunay triangulations and Voronoi diagrams.

The No-Certificate Scenario is likely to occur in various application contexts, such as CAD models, where 3D subdivisions are often represented with no topological information [11]). Also, the availability of N-checkers is important when one tries to incorporate in a program modules developed by others, whose source code may be difficult to understand or modify.

This paper contains a systematic analysis of checkers for various types of planar subdivisions and for convex polytopes in two and higher dimensions. We advocate a new requirement that good geometric checkers should satisfy, and present simple and efficient N-checkers for several structures for which only A-checkers and T-checkers have been so far designed. Specifically, our contributions can be summarized as follows.

1. As an additional measure of effectiveness for a checker, we adopt the notion of *degree* [16, 17, 5], which takes into account the number of bits required by the checker to carry out error-free computations. A good checker should have degree no higher than the problem at hand allows. We give lower bounds on the degree of checkers for planar subdivisions and convex polytopes, and present optimal-degree checkers.

2. We present a new T-checker for convex polytopes that is simpler than the one given in [19]. Our T-checker works in any dimension and recursively reduces the verification of a d-dimensional polytope to the verification of an associated $(d-1)$-dimensional polytope. The T-checker is optimal with respect to both the time complexity and the degree. The design of our T-checker for convex polytopes reveals new combinatorial and geometric properties that may be of independent interest.

3. We present linear-time optimal-degree T-checkers for triangulations, convex subdivisions, and general planar subdivisions. Such checkers use as subroutines elementary graph algorithms and do not require to test the planarity of the underlying input graph.

4. Extending the above results on T-checkers, we present linear-time optimal-degree N-checkers for triangulations and convex subdivisions. This solves significant special cases of an open problem mentioned by Kirkpatrick [14] on the existence of an $o(n \log n)$ algorithm to verify the planarity of a geometric graph.

5. As a further application, we give linear-time optimal-degree N-checkers for Delaunay triangulations, locally-minimum-weight triangulations and Delaunay diagrams.

Near the completion of our investigations, we became aware of two ongoing projects on the design of T-checkers for planar subdivisions, including triangulations and convex subdivisions. A manual [18] describing the functionality of C++ functions that implement T-checkers for Delaunay triangulations, Voronoi diagrams and convex planar subdivisions is available from Mehlhorn's Web page. A manuscript in progress [20] contains characterizations of triangulations and convex planar subdivisions similar to those of the present paper. There is therefore an undeniable shared objective between our research and that of Mehlhorn *et al.* Any minor overlaps of independently obtained results are largely offset by the fact that the respective proof techniques are different, and that our checkers work also within the No-Certificate Scenario, while maintaining the same efficiency as the T-checkers of [20, 18].

The rest of this paper is organized as follows. Preliminaries and lower bounds on the degree of the problems studied in this paper are given in Section 2. Our T-checker for convex polytopes is presented in Section 3. Section 4 is devoted to T-checkers for triangulations, convex subdivisions, and general planar subdivisions. N-checkers are studied in Section 5. Finally open problems are given in Section 6.

2 Preliminaries

We start with definitions for geometric graphs, ordered graphs, and planarity. We then recall the notion of degree of geometric algorithms, introduced in [16, 17] (a related concept is defined in [5]). Finally, we present lower bounds on the degree of checkers for convex polytopes and planar subdivisions.

2.1 Geometric graphs, ordered graphs, and planarity

A d-dimensional *geometric graph* is a graph drawn with straight-line edges in d-dimensional space, i.e., a graph whose vertices have d-dimensional coordinates. In the following, we often denote with Γ a geometric graph, and with G its underlying combinatorial structure. To simplify the notation, and when the context is unambiguous, we may denote with G both the geometric graph and its underlying combinatorial structure.

A two-dimensional geometric graph Γ is *planar* if it has no crossing edges, i.e., any two edges of Γ intersect only at a common vertex. For every planar graph G, there exists a planar geometric graph Γ with underlying graph G, i.e., every planar graph admits a planar straight-line drawing (see, e.g., [10]). However, a geometric graph with an underlying planar graph is not necessarily planar.

A planar geometric Γ graph determines a *planar subdivision* S, i.e., a partition of the plane into regions called faces. Planar subdivision S is said to be *induced* by Γ. A planar subdivision is *convex* if the boundary of each face is a convex polygon. A planar subdivision S induced by geometric graph Γ is *maximal* if there is no other planar geometric graph Γ' such that Γ is a subgraph of Γ'. In a maximal planar subdivision, the boundary of each internal face is a triangle, and the boundary of the external face is a convex polygon. A maximal planar subdivision is also called a *triangulation*.

A graph G is *ordered* (or G has an *ordering*) if for each vertex v of G, a circular ordering of the edges incident on v is given. The ordering of a graph is usually denoted with Ψ. A two-dimensional geometric graph Γ has a natural ordering associated with it, called *the ordering of* Γ, given by the clockwise circular sequence of the edges incident on each vertex.

An ordering Ψ of a graph induces a set of directed *circuits*, where the edge (v, w) following (u, v) in a circuit is the successor of (u, v) in the circular ordering of the edges incident on v. Every edge of the graph is traversed by exactly two circuits, once in each direction.

Let Γ be a geometric graph, let Ψ be its ordering, and let v be a vertex of Γ with largest y-coordinate. Let (u, v) and (v, w) be two edges of Γ incident on v such that the wedge defined by two rays emanating from v and passing through u and w contains all other edges incident on v. We call *outer circuit* of Γ the circuit induced by Ψ that contains edges (u, v) and (v, w). The remaining circuits induced by Ψ are said to be *internal circuits*.

If Γ is planar, then the circuits induced by Ψ are the boundaries of the faces of the subdivision S induced by Γ. In particular, the outer circuit is the boundary of the external face traversed clockwise, and the internal circuits are the boundaries of the internal faces traversed counterclockwise.

The ordering Ψ of an ordered graph G is *planar* if there exists a planar geometric graph Γ whose underlying graph is G and whose ordering is Ψ. A planar ordering of G is associated with a planar (topological) embedding of G. A graph G admits a planar ordering only if it is planar. Also, the number of distinct planar orderings (topological embeddings) of a planar graph can be super-exponential.

2.2 Degree of geometric algorithms and problems

The numerical computations of a geometric algorithm are basically of two types: tests (predicates) and constructions. The two types of computations have clearly distinct roles. Tests are associated with branching decisions in the algorithm that determine the flow of control, whereas constructions are needed to produce the output data of the algorithm.

Approximations in the execution of constructions are acceptable, since approximate results are perfectly tolerable, provided that the error magnitude does not exceed the resolution required by the application. On the other hand, approximations in the execution of tests may produce an incorrect branching of the algorithm. Such event may have catastrophic consequences, giving rise to *structurally* incorrect results. The exact-computation paradigm therefore requires that tests be executed with total accuracy.

We therefore characterize geometric algorithms on the basis of the complexity of their test computations. Any such computation consists of evaluating the sign of an algebraic expression over the input variables, constructed using an adequate set of operators, such has $\{+, -, \times, \div, \sqrt[2]{}\}$. This can be reduced to the evaluation of the signs of multivariate polynomials derived from the expression.

We make here the reasonable assumption that input data be dimensionally consistent, i.e., that, if an entity with the physical dimension of a length is represented with b bits, then one with the dimension of an area be represented with $2b$ bits, and so on. Under the hypothesis of dimensional consistency (where point coordinates are b-bit entries), a polynomial expressing a test is homogeneous because all of its monomials must have the same physical dimension.

A primitive variable is an input variable of the algorithm and has conventional arithmetic degree 1. The arithmetic degree of a polynomial expression

E is the common arithmetic degree of its monomials. The arithmetic degree of a monomial is the sum of the arithmetic degrees of its variables. An algorithm has *degree* d if its test computations involve the evaluation of multivariate polynomials of arithmetic degree at most d [16, 17]. A problem Π has *degree* d if any algorithm that solves Π has degree at least d.

Theorem 1 *[16, 17] The degree of the problem of evaluating a predicate expressed by a polynomial P is the maximum arithmetic degree of the irreducible factors of P that change sign over their domain.*

We now state lower bounds on the degree of the checkers that will be studied in the rest of the paper.

A 3-*dimensional polytope* is the (convex) 3-dimensional intersection of 3-dimensional half-spaces.

A *simplicial* 3-*dimensional polyhedron* Γ is a 3-dimensional geometric graph with a planar ordering that induces faces that are 3-cycles. The faces induced by the planar ordering correspond to the facets of Γ. Γ is *convex* if it is the boundary of a convex polytope. It is *locally convex* if for each edge (p_1, p_2) $(p_i = (x_i, y_i, z_i))$, with incident facets $p_1p_2p_3$ and $p_1p_2p_4$, the simplex (p_1, p_2, p_3, p_4) lies in the interior half-space of each of the two planes supporting the facets. We assume that facets of 3-dimensional polyhedra and edges of 2-dimensional polygons are open sets. A *simplicial d-dimensional polyhedron* is a simplicial $(d-1)$-dimensional surface without boundary. A *d-dimensional polytope* is the (convex) d-dimensional intersection of d-dimensional half-spaces.

An A- or a T-checker for 3-dimensional convex polytopes receives as input a simplicial 3-dimensional polyhedron, and verifies its convexity. An N-checker for 3-dimensional convex polytopes receives as input 3-dimensional geometric graph, and verifies whether such graph is a convex simplicial 3-dimensional polyhedron. The input an A- , T-, and N-checker for d-dimensional $(d > 3)$ convex polytopes is analogously defined.

In the next theorems we use the generic term "checker" to denote either an A-, or a T-, or an N-checker.

Theorem 2 *A checker for* 3*-dimensional convex polytopes has degree at least* 3*.*

Proof: Let Γ be an input of the checker, such that Γ is isomorphic to K_4. Let $(p_i = (x_i, y_i, z_i))$ $(i = 1, \ldots, 4)$ the four vertices of Γ. Γ is globally convex if and only if its four vertices are not co-planar. This is equivalent to specifying that the determinant Δ_{1234} defined below be non-zero.

$$\Delta_{1234} = \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix}$$

Since points p_1, p_2, p_3 and p_4 are independent, Δ_{1234} is an irreducible polynomial over the rationals. Hence, by Theorem 1, the problem has degree 3. $\diamondsuit$

In the rest of this paper we will refer to the geometric primitive that determines the sign of a determinant of the type Δ_{1234} as an *orientation test*. Referring to Figure 1, we observe that testing the condition of local convexity at edge (p_1, p_2) corresponds to determining whether the sign of Δ_{1234} is nonnegative. Hence, also a local convexity test corresponds to an orientation test. We use the notation $|p_1, p_2, p_3, p_4|$ to denote the determinant whose i-th row is $[x_i, y_i, z_i, 1]$ $(i = 1, \ldots, 4)$; $|p_1, p_2, p_3|$ is the determinant whose i-th row is $[x_i, y_i, 1]$ $(i = 1, \ldots, 4)$.

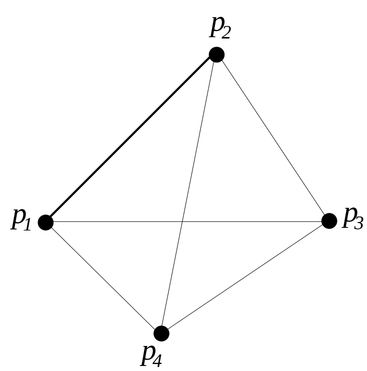

Figure 1: Simplex (p_1, p_2, p_3, p_4). The local convexity test for edge (p_1, p_2) (highlighted in the picture) corresponds to the orientation test $|p_1, p_2, p_3, p_4|$.

With analogous reasoning as in the proof of Theorem 2, the following lower bounds can be established.

Theorem 3 *A checker for 3-dimensional convex polytopes has degree at least d.*

An A- or a T-checker for 2-dimensional planar subdivisions receives as input an ordered 2-dimensional geometric graph with a given ordering, and verifies whether any two edges cross. An N-checker for 2-dimensional planar subdivision receives as input a 2-dimensional geometric graph, computes the ordering of the graph and verifies whether any two edges cross.

Theorem 4 *A checker for* 2*-dimensional planar subdivision has degree at least* 2*.*

3 T-checkers for Convex Polytopes

In this section, we describe the design of a T-checker for convex polytopes in a fixed dimension d. The input to the checker is a simplicial polyhedron Γ in E^d. The planar ordering of Γ serves as a certificate describing the topology of the polytope claimed to be induced by Γ.

Because of its appeal to intuition, we consider the question for $d \leq 3$ (specifically for $d = 2$ and $d = 3$) and indicate later how to extend the result to higher dimensions. In Subsections 3.2 and 3.3 we simplify the notation and use the term polyhedron to denote a simplicial 3-dimensional *polyhedron* and the term *polytope* to denote 3-dimensional polytope.

3.1 Previous Results

Let Γ be a locally convex polygon. Given an edge e of Γ, the line through e divides the plane into two half-planes. The closed half-plane containing the two edges incident on e is said to be the *negative* half-plane of e. The other half-space is said to be *positive*. The *core* of Γ, denoted by $\chi(\Gamma)$, is the interior of the convex polygon obtained as the intersection of all negative half-planes defined by the edges of Γ. Note that the above definition of core is different from that of the kernel of a simple (i.e., not intersecting) polygon.

Given a facet f of a locally convex polyhedron Γ, the plane supporting f divides the space into two half-spaces. We call *negative* the half-space that contains the facets adjacent to f. The *core* of Γ, denoted by $\chi(\Gamma)$, is the interior of the convex polytope $\chi(\Gamma)$ obtained as the intersection of all negative half-spaces defined by the facets of Γ.

Mehlhorn et al. [19] have proved results equivalent to those stated in the theorems and lemmas that follow.

Lemma 1 [19] *Let Γ be a locally convex polygon and let q be any point of $\chi(\Gamma)$. Every ray emanating from q intersects the same number of edges and/or vertices of Γ.*

Theorem 5 [19] *A locally convex polygon* Γ *is globally convex if and only if a ray emanating from a point of* $\chi(\Gamma)$ *intersects* Γ *at a single edge or at a single vertex.*

Lemma 2 [19] *Let* Γ *be a locally convex polyhedron and let* q *be any point of* $\chi(\Gamma)$. *Every ray emanating from* q *and intersecting* Γ *only at facets intersects the same number of facets of* Γ.

Theorem 6 [19] *A locally convex polyhedron* Γ *is the boundary of a convex polytope if and only if any ray emanating from a point of* $\chi(\Gamma)$ *and intersecting a facet of* Γ *does not have any other intersection with* Γ.

Based on Theorem 6, Mehlhorn et al. [19] check whether a locally convex polyhedron is the boundary of a convex polytope by first computing a point q of $\chi(\Gamma)$ and a ray r emanating from from q and passing through the centroid of an arbitrarily chosen facet, and then checking that no other facet is intersected by r. Clearly, this checker runs in time linear in the number of vertices, and is therefore optimal in the conventional asymptotic sense. Moreover, it uses the minimum degree (3, for the orientation test).

We now discuss an equivalent criterion that is also degree-optimal and simpler in the sense that its linear-time performance is characterized by a lower constant. This feature is particularly significant for a checker, where "practical" speed is at a premium, and reflects an emerging philosophy on the inadequacy of asymptotic analysis. We will prove the validity of our criterion by showing its equivalence to the criterion of Mehlhorn et al. (Theorem 6).

3.2 A New Convexity Criterion

We start by characterizing the global convexity of a locally convex polygon. Let Γ be a locally convex polygon, given as a circular sequence of vertices. A vertex v of Γ is said to be a *2-seam vertex* of Γ if:

- the intersection of the negative half-planes defined by the two edges sharing v is contained in one of the closed half-planes defined by the vertical (i.e., parallel to the y-axis) line ℓ through v, and
- the upper (according to the positive y-direction) edge incident on v is contained in one of the open half-planes defined by ℓ.

The 2-*seam* of Γ is the set of its 2-seam vertices. The following theorem is straightforward.

Theorem 7 *A locally convex polygon is globally convex if and only if its 2-seam consists of two vertices.*

We now discuss the more complicate case of T-checking the convexity of a (3-dimensional) polyhedron. Let Γ be a locally convex polyhedron.

An edge e of Γ is said to be a *seam edge* of Γ if:

- e is not vertical,
- the intersection of the negative half-spaces defined by the two facets sharing e is contained in one of the closed half-spaces defined by the vertical (i.e., parallel to the z-axis) plane Π through e, and
- the upper (according to the positive z-direction) facet incident on e is contained in one of the open half-spaces defined by Π.

The *seam* of Γ, denoted by $\sigma(\Gamma)$, is the subgraph of Γ induced by its seam edges.

Lemma 3 *For each vertex v of $\sigma(\Gamma)$ there are at least two seam edges incident on v.*

Proof: We shall constructively show that, given a seam edge (u, v), we can find another seam edge (v, u'). Consider a sphere S whose center is v and whose radius is small enough so that any other vertex of Γ and edge of Γ not incident to v is outside S. The intersection between S and Γ defines a spherical polygon P on S.

The local convexity of Γ induces local convexity on P. We observe that seam edges of Γ incident to v correspond to vertices of P that have a supporting meridian (i.e., a great circle in a vertical plane). Hence, denoting by w be the vertex of P defined as the intersection point between (u, v) and S, we have that w has a supporting meridian. Thus, starting at w it suffices to follow the sequence of edges on P to find another vertex w' of P with supporting meridian, which corresponds to a new seam edge (v, u'). $\diamondsuit$

We say that the seam $\sigma(\Gamma)$ is *degenerate* if it has a vertex with more than two seam edges incident on it. Figure 2 shows a degenerate seam.

Lemma 4 *If $\sigma(\Gamma)$ is degenerate, then Γ is not globally convex.*

Proof: Let e_1, e_2, and e_3 be three seam edges incident on the same vertex v. Edges e_1, e_2, and e_3 have three vertical supporting planes Π_1, Π_2, and

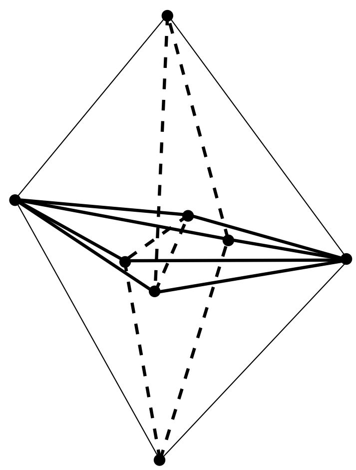

Figure 2: A locally convex polygon that has a degenerate seam. The seam edges are drawn with thick lines.

Π_3, respectively. Let W be the common intersection of the negative half-spaces defined by Π_1, Π_2, and Π_3. Without loss of generality, let Π_1 and Π_2 be the planes delimiting W. In order to be globally convex, Γ must be inside W. However, Π_3 is not internal to W. If it is external to W, so is e_3; if it coincides with, say, Π_1, then e_3 is still external to W, otherwise it would violate the seam edge definition. In either case, we conclude that Γ is not globally convex. $\diamondsuit$

The following lemma reveals the combinatorial structure of the seam $\sigma(\Gamma)$. In the lemma, the term disjoint cycles has to be intended in the graph sense.

Lemma 5 *If $\sigma(\Gamma)$ is nondegenerate, then it is a collection of disjoint cycles.*

Proof: Since $\sigma(\Gamma)$ is nondegenerate, every vertex of $\sigma(\Gamma)$ has at most two incident seam edges by definition. By Lemma 3, every vertex of $\sigma(\Gamma)$ has at least two incident seam edges. $\diamondsuit$

By *z-projection* of a geometric object o (point, edge, face) we mean its projection parallel to the z-axis and we denote it as o'.

Lemma 6 *Let $\sigma(\Gamma)$ be nondegenerate, let q be a point of $\chi(\Gamma)$, let r be any nonvertical ray emanating from q, and let r' be the z-projection of r. If the intersection of r with Γ is at k facets, then the intersection of r' with the z-projection $\sigma'(\Gamma)$ of $\sigma(\Gamma)$ is at k edges and/or vertices.*

Proof: Assume that the intersection of ray r with Γ is at the k facets $f_1, \ldots, f_k$, $k \geq 2$. Consider the vertical plane α containing r, and let $\alpha^\star \subset \alpha$ be the half-plane containing r and limited by the vertical line ℓ through q.

Let p_i be the intersection of r with f_i. Points p_i $(i = 1, \ldots, k)$ belong to a closed curve in α (since Γ is a surface without boundary). Since Γ is a locally convex polyhedron, such closed curve in α is a locally convex polygon and ℓ traverses the core of such polygon. As a consequence, each point p_i belongs to a distinct polygonal chain in $\alpha^\star$ (see Figure 3 for the case $k = 2$).

Starting from p_i and proceeding on the corresponding polygonal chain by increasing z we reach a point of maximum z in $\alpha^\star$, denoted t_i (either t_i is a local maximum in α, or t_i is on ℓ). Analogously, if we proceed from p_i by decreasing z we will reach a point of minimum z in $\alpha^\star$, denoted b_i. If t_i is not a point of ℓ, then it admits a horizontal supporting line in α; otherwise, the angle δ formed by the ascending polygonal line at t_i with the vertical by q is $\delta > \pi/2$ (see Figure 3). Analogously, b_i either admits a horizontal supporting line or the angle of the corresponding angle is $\delta' < \pi/2$. We conclude that traversing downwards the subchain γ_i from t_i to b_i we reach a point s_i which is the first to admit a vertical supporting line in $\alpha^\star$, i.e., s_i belongs to the seam. Observe that, because of the local convexity of Γ, there is no other point on γ_i having a supporting vertical line.

Thus, for each γ_i in $\alpha^\star$ intersected by r there is a unique point s_i that belongs to the seam. Hence, since intersection is preserved by projection, each s_i projects to a point on the z-projection r' of r. Also, because $\sigma(\Gamma)$ is nondegenerate, it cannot happen that two distinct s_i and s_j project to the same point on r', which proves the lemma. $\diamondsuit$

The above results allow us to derive a new characterization of convex polytopes and reduce by one the dimensionality of the criterion given by Theorem 6.

Theorem 8 *A 3-dimensional simplicial polyhedron Γ is the surface of a convex polytope if and only if the following two conditions hold:*

1. *Γ is locally convex,*

2. *$\sigma'(\Gamma)$ is a globally convex polygon.*

Proof: We first show that if Γ is globally convex, then Conditions 1, and 2 are satisfied. Clearly, if Γ is globally convex, it is also locally convex. Also, the z-projection of a globally convex polyhedron is globally convex and coincides with the z-projection of its seam.

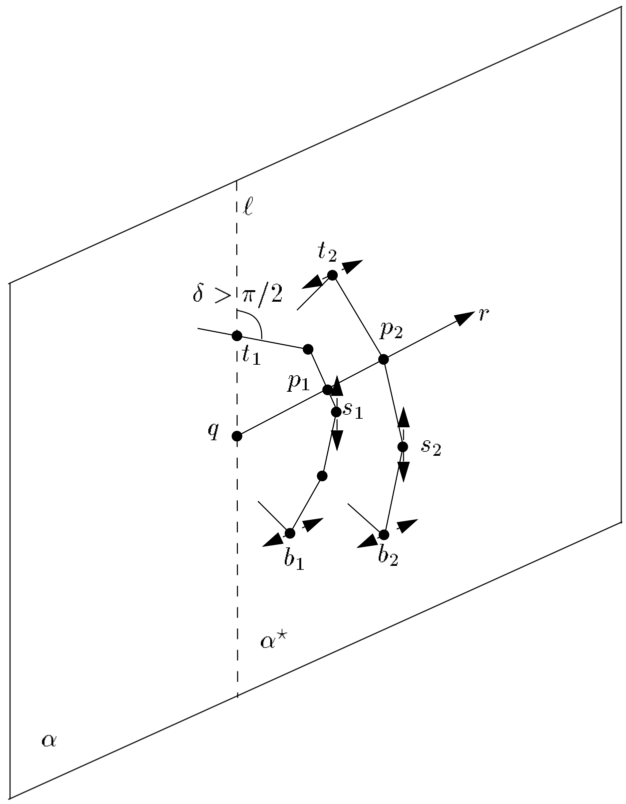

Figure 3: Example of cross section in plane $\alpha^\star$ for the proof of Lemma 6.

We now prove that if Conditions 1, and 2 are satisfied, then Γ is globally convex. By Condition 2, and because of Lemma 5, we have that $\sigma(\Gamma)$ is nondegenerate. Also, since $\sigma'(\Gamma)$ is a globally convex polygon, its core coincides with the polygon itself, i.e. $\chi(\sigma'(\Gamma)) = \sigma'(\Gamma)$.

We begin by showing that the core of Γ, $\chi(\Gamma)$, is nonempty. Take any point q' of $\chi(\sigma'(\Gamma))$ and consider a vertical plane Π through q'. The intersection between Π and Γ is a closed curve, since Γ is a surface without boundary. Let γ be such a curve. Observe that γ is a locally convex polygon since, by Condition 1, Γ is locally convex. Also, each 2-seam vertex of γ corresponds to a seam edge of Γ that is vertically projected to $\sigma'(\Gamma)$. Since Π intersects $\sigma'(\Gamma)$ at two edges and/or vertices, we have that γ can have only two 2-seam vertices. Thus, by Theorem 7, γ is a globally convex polygon. Hence, a vertical line ℓ through q' intersects γ only twice. We take any point along ℓ and in the interior of polygon γ and call it q. Observe that rotating the vertical plane Π around ℓ, one can define infinitely many globally convex polygons, given by the intersection of the rotated plane and Γ. Also notice that q lies in the interior of any of such globally convex polygons. We can conclude that q is on the negative side of all facets of Γ, and thus it belongs to $\chi(\Gamma)$.

We are now in the position of show that Γ is globally convex. Consider point q and a nonvertical ray r emanating from q such that r intersects Γ only at facets (clearly, there always exists such a ray). Let k be the number of facets of Γ intersected by r. We prove that $k = 1$ and thus, by Theorem 6, Γ is globally convex.

Let r' be the z-projection of r. Ray r' originates from point q' that is a point of $\chi(\sigma'(\Gamma))$ by construction. By Lemma 6, the intersection of r' with the z-projection $\sigma'(\Gamma)$ of $\sigma(\Gamma)$ consists of k edges and/or vertices. Since $\sigma'(\Gamma)$ is globally convex, by Lemma 1 every ray emanating from q' intersects $\sigma'(\Gamma)$ the same number of times. Also, by Theorem 5, r' intersects $\sigma'(\Gamma)$ at exactly one edge and/or vertex, that is $k = 1$. $\diamondsuit$

3.3 Checking Algorithm

Theorem 8 allows us to check whether a 3-dimensional simplicial locally convex polyhedron is the boundary of a convex polytope by testing the z-projection of its seam $\sigma'(\Gamma)$ for global convexity using Theorem 7.

The pseudocode of our T-checker for 3-dimensional convex polytopes is given in Fig. 4.

Lemma 7 *Algorithm T-check-polytope correctly checks whether a polyhedron is the boundary of a convex polytope.*

Proof: The certificate describing the topology is used in the local convexity test. It identifies the seam edges, and the two vertices of each such edges are tested for memberships in the 2-seam. Since, among the vertices of the 2-seam there are identical numbers of x-minimum vertices and x-maximum vertices, it suffices to verify that there is only one x-maximum vertex.

The algorithm examines each vertex that is the endpoint of a seam edge and counts the number of seam edges incident on it by means of variable $d(v)$. If $d(v) > 2$ then the seam is degenerate and, by Lemma 4, Algorithm T-check-polytope concludes that the input is not the boundary of a convex polytope.

Algorithm *T-check polytope*

```
r ← 0
foreach vertex v ∈ Γ do
  d(v) ← 0
foreach edge e ∈ Γ do
  if e is not locally convex
    then return false
    else if e is a seam edge
      then let w[0] and w[1] be the two endpoints of e
        for i ← 0, 1 do
          v ← w[i]
          u ← w[(i + 1) mod 2]
          if d(v) = 2
           then return false
           else if d(v) = 0
             then d(v) ← 1
               p(v) ← u
             else { d(v) = 1 }
               d(v) ← 2
               if v is a right-2-seam vertex (based on p(v) and u)
                 then if r = 0
                   then r ← 1
                   else return false
return true
```

Figure 4: T-checker for 3-dimensional convex polyhedra.

Otherwise, the algorithm verifies whether the z-projection of v belongs to the 2-seam and has a local maximum abscissa, by comparing the x-coordinate of v with that of the other two endpoints u and $p(v)$ of the seam edges incident on v. Namely, if the x-value of v is larger than the x-values of both u and $p(v)$, then we say that v belongs to the *right-2-seam.* If the x-value of v is equal to the abscissa of one of its neighbors in the seam, say $p(v)$, and larger the abscissa of u, then v belongs to the right-2-seam only if the y-value of v is larger or equal to the y-value of $p(v)$. Otherwise, v does not belong to the right-2-seam.

When a vertex is determined to be in the right-2-seam if the variable r has value 0, then it is set to 1. Otherwise, if the value of r is found to be already 1 (which means that there are at least two vertices in the right-2-seam), Algorithm T-check-polytope concludes that the input is not the boundary of a convex polytope. $\diamondsuit$

We now discuss a low-degree implementation of **Algorithm** *T-check polytope*.

We observe that the test for membership of an edge in the seam is a subcomputation of its test for local convexity. Namely, let $\Delta_{1234} > 0$ be the condition for local convexity on edge $(p_1 p_2)$. (See Figure 5.)

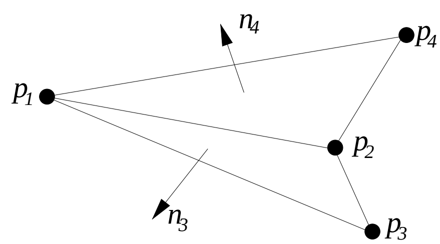

Figure 5: The outward normals n_4 and n_3.

The outward normal n_4 to $p_1 p_2 p_4$ has $z-$component

$$\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_4 - x_1 & y_4 - y_1 \end{vmatrix} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} = \Delta_{124}$$

and n_3 has $z-$component

$$-\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = -\Delta_{123}$$

Therefore, if we evaluate Δ_{1234} by expansion according to the $z-$column, we obtain (free) the minors required to test vertical support, i.e. $sign(\Delta_{124} \cdot \Delta_{123}) = \{0, +\}$.

We can summarize the above discussion as follows.

Theorem 9 *Let Γ be a locally convex polyhedron with n vertices in 3-dimensional space. There exists a T-checker that verifies whether Γ is the boundary of a convex polytope in $O(n)$-time and with optimal-degree 3. Furthermore, the T-checker evaluates at most $3n - 6$ predicates of degree 3.*

We observe that the checker for convex polytopes given by Mehlhorn et al. [19] evaluates in the worst case $6n-15$ predicates of degree 3. Namely, their checker verifies first the local convexity condition at each edge. This implies executing $3n-6$ orientation tests of degree 3. Then, the centroid o of the polyhedron and the centroid p of a facet are computed and it is verified whether the ray emanating from o and passing through p does not have any other intersection with Γ. To this end, a degree 3 orientation test for each edge of Γ except the three edges of f is necessary, thus giving rise to additional $3n-9$ orientation tests.

3.4 Extension to d-dimensional Space

The approach described in the previous sections can be generalized to higher dimensions. Let Γ be a simplicial polyhedron in d dimensions. A j-facet of Γ is a simplex defined by j linearly independent points; a conventional facet is a d-facet. Given points $p_1, \ldots, p_j$ we let $[p_1 \ldots p_j]$ denote the $j \times (d+1)$ matrix whose i-th row is $(x_1^{(i)}, \ldots, x_d^{(i)}, 1)$; $||p_1 \ldots p_j||$ is the determinant of the leftmost j columns of $[p_1 \ldots p_j]$.

The initial local convexity test applies to a $(d-1)$-facet. Specifically, let $(d-1)$-facet F be defined by $p_1, \ldots, p_{d-1}$, indexed so that $||p_1 \ldots p_{d-1}|| > 0$. F is shared by two d-facets respectively defined by $p_1, \ldots, p_{d-1}, p_d$ and $p_1, \ldots, p_{d-1}, p_{d+1}$, for two additionally independent points p_d and p_{d+1}. We say that F is *locally convex* if $||p_1 \ldots p_{d-1} p_d|| > 0$ and $||p_1 \ldots p_{d-1} p_d p_{d+1}|| > 0$. We also say that f belongs to the d-seam if $||p_1 \ldots p_{d-1} p_d|| > 0$ and $||p_1 \ldots p_{d-1} p_{d+1}|| \leq 0$. A $(d-1)$-facet in the d-seam is easily shown to belong to a hyperplane parallel to the x_d-axis. If we project all d-seam $(d-1)$-facets along the x_d-axis we obtain a $(d-1)$-dimensional polytope Γ', which can be shown to inherit the property of local convexity. The seam test for Γ' now involves membership of $(d-2)$-facets in a $(d-1)$-seam. Each $(d-1)$-facet F in the d-seam shares a $(d-2)$-facet with $(d-1)$ adjacent $(d-1)$-facets, all belonging to the d-seam. Let F' be one such $(d-2)$-facet (for $d = 3$ F' is a point), which is visited twice as all d-seam facets are visited; through the obvious use of a pointer the two sharing $(d-1)$-facets are linked to each other, and F' can be tested for membership in the $(d-1)$-seam; if F' is visited more than twice, then Γ can be rejected as nonconvex. Since the correct orientation of a single simplex is equivalent to the evaluation of a nonfactorizable determinant of degree d, d is a lower bound on the needed degree.

The results presented in this section are summarized in the following theorem.

Theorem 10 *Let Γ be a d-dimensional polyhedron for a fixed dimension d. There exists a T-checker that verifies whether Γ is the boundary with optimal-degree d.*

4 T-checkers for Planar Subdivisions

In this section we consider T-checkers for planar subdivisions that take as input a two-dimensional geometric graph Γ and the ordering Ψ of Γ, and verify whether Γ induces a planar subdivision satisfying a certain predicate (e.g., convex faces).

A building block of such a T-checker is an algorithm that tests whether an ordering Ψ of a graph G is planar. A linear-time algorithm for answering the question was given by Kirkpatrick in [14]. His algorithm considers the circuits induced by Ψ and adds to G a new vertex v_c for each induced circuit c and a new edge (v_c, w) for each vertex w of c. The resulting augmented graph G^* is planar if and only if the ordering Ψ is planar. Thus, the planarity of Ψ can be checked by running a planarity-testing algorithm (e.g., [13]) on G^*. Besides its theoretical interest, however, this algorithm may be not be the most suited for practical applications, since the implementation of a linear-time planarity testing algorithm is complex and requires sophisticated data structures.

We show that there exists a much simpler solution to the problem of testing in linear time whether an ordering Ψ of a connected graph G is planar. Our algorithm follows from basic results in planarity theory [12]. Namely, we determine the circuits induced by Ψ and check whether their number C is equal to $E - V + 2$, where V and E are the number of vertices and edges of G, respectively. Clearly, this takes linear time.

Lemma 8 *Let Γ be a two-dimensional connected geometric graph with a planar ordering Ψ. If Γ is not planar (i.e., it has crossing edges), then at least one circuit of Γ induced by Ψ is not planar (i.e., it is a self-intersecting polygon).*

Proof: Let G be the graph underlying G. Ordering Ψ induces a planar embedding of G whose faces are the circuits of Γ induced by Ψ. The proof is by induction on the number of faces of G (i.e., circuits of Γ). Clearly, if G has only one face f (f is in the case the external face) and Γ is not planar, then the induced circuit of Γ associated with f is a self-intersecting polygon.

Suppose now that the lemma holds for k faces and that G has $k+1$ faces. Let f be an internal face associated with a circuit c of Γ that shares one or more edges with the outer circuit. If c is self-intersecting, then we are done. Otherwise, remove the edges of c from the outer circuit of Γ, and let Γ' be the resulting geometric graph. If Γ' is not planar, then Γ', and thus also Γ, has a self-intersecting circuit by the inductive hypothesis. Otherwise, the outer circuit of Γ is a self-intersecting circuit. $\diamondsuit$

By Lemma 8, verifying whether a connected geometric graph Γ with a planar ordering is a planar can be reduced to testing separately whether the circuits induced by Ψ are simple (i.e., not self-intersecting) polygons. Lemma 8 also provides an alternative and shorter proof of a result of [9] (Lemma 4.5), as the following corollary shows.

Corollary 1 *Let Γ be a connected two-dimensional geometric graph. If the ordering Ψ of Γ is planar, all the internal circuits induced by Ψ are triangles, and the outer circuit induced by Ψ is a convex polygon, then Γ is planar and induces a triangulation.*

Observe, however, that Corollary 1 cannot be easily extended to other types of planar subdivisions. For example, Figure 6 shows a triconnected geometric graph (obtained by removing two edges from a triangulation) that has a planar ordering but is itself not planar.

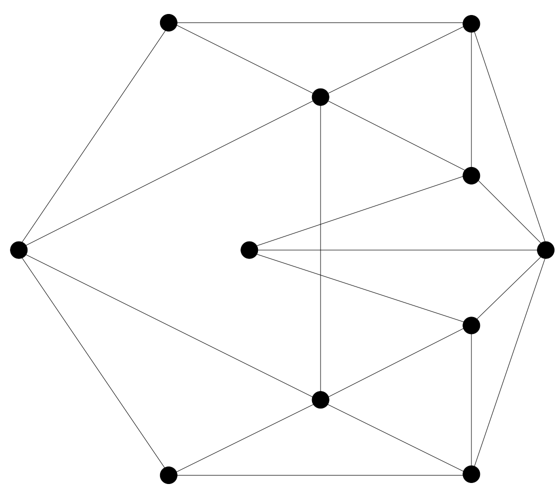

Figure 6: A triconnected geometric graph whose ordering is planar, but which is itself not planar.

Lemma 9 *A connected two-dimensional geometric graph Γ with ordering Ψ is planar and induces a triangulation if and only if:*

1. *ordering Ψ is planar;*

2. *all the internal circuits induced by Ψ are triangles; and*

3. *the outer circuit induced by Ψ is a convex polygon.*

Proof: Follows from Lemma 8 and from the fact that no triangle can be a self-intersecting polygon. $\diamondsuit$

The input to our T-checker for triangulations is a two-dimensional geometric graph Γ with n vertices and the ordering Ψ of Γ. The T-checker constructs the circuits induced by Ψ and then performs the following sequence of 6 simple checks. It fails as soon as one of the checks fails, and succeeds if all the checks succeed. Checks 1 and 6 have degree 2. The other checks have degree 1.

1. Check that Ψ is the ordering of Γ.

2. Check that ordering Ψ is planar.

3. Check that Γ is connected.

4. Check that Γ has no more than $3n - 6$ vertices.

5. Check that all internal circuits induced by Ψ are triangles.

6. Check that the outer circuit induced by Ψ is a convex polygon.

Theorem 11 *There exists an optimal T-checker for planar triangulations that runs in linear time and has degree* 2.

Lemma 9 can be generalized to convex planar subdivisions.

Lemma 10 *A two-dimensional connected geometric graph Γ with ordering Ψ is planar and induces a convex planar subdivision if and only if:*

1. *ordering Ψ is planar; and*

2. *all the circuits induced by Ψ are convex polygons.*

Theorem 12 *There exists an optimal T-checker for convex planar subdivisions that runs in linear time and has optimal degree* 2.

As shown in Fig. 6, there exist nonplanar geometric graphs with a planar ordering and an underlying triconnected graph. In order to verify whether such geometric graphs are planar, the strategy suggested by Lemma 8 is to check the simplicity of the circuits induced by the ordering are simple polygons. Testing the simplicity of a polygon with k vertices can be done in $O(k \log k)$ by a simple optimal-degree sweep-line algorithm [3] or in $O(k)$-time as an application of the more elaborate triangulation algorithm by Chazelle [6]. This discussion can be summarized as follows.

Lemma 11 *A connected geometric graph Γ with ordering Ψ is planar (and thus induces a planar subdivision) if and only if:*

1. *ordering Ψ is planar; and*
2. *all the circuits induced by Ψ are simple polygons.*

Theorem 13 *There exists a T-checker for planar subdivisions that runs in $O(n \log n)$ time and has degree 2, where n is the number of vertices.*

5 N-checkers for Planar Subdivisions

In this section, we present N-checkers for planar subdivisions. Our N-checkers compute the ordering of the input geometric graph and then use one of the T-checkers presented in Section 4.

Theorem 14 **[14]** *Let Γ be a geometric graph with n vertices such that its underlying graph G is planar and has $\lambda(G)$ distinct planar orderings. There exists an algorithm that either computes the ordering of Γ in $O(n+\log \lambda(G))$ time or fails. If it fails, then the ordering of Γ is not planar.*

By a detailed analysis of the algorithm presented in [14] we obtain:

Lemma 12 *The algorithm of Theorem 14 has degree 2.*

Our N-checker for planar triangulations exploits Theorem 14, Lemma 12, and the fact that the underlying graph G of a triangulation has $\lambda(G) = 2$. Let Γ be a geometric graph with n vertices. By Theorem 14, we can compute the ordering of Γ in $O(n)$ time or else we can conclude that Γ is not a triangulation (we reach this conclusion either when the running time of Kirkpatrick's algorithm exceeds $O(n)$ or when Kirkpatrick's algorithm fails in $O(n)$-time). Once the ordering of Γ is computed, we apply the T-checker of Theorem 11.

Theorem 15 *There exists an optimal N-checker for planar triangulations that runs in linear time and has degree* 2*.*

Theorem 16 *There exists an N-checker for Delaunay triangulations that runs in linear time and has degree* 4*.*

Proof: First, we verify that Γ is planar and induces a triangulation S using the T-checker of Theorem 15. To verify that S is a Delaunay triangulation, it suffices to check whether for every triangle $T = \triangle(a,b,c)$ of S, the disk through a, b, c contains any of the opposite vertices in the triangles sharing one edge with T. Clearly, this can be done in linear time. Also, the above in-circle test can be executed with a degree 4 algorithm (see, e.g. [16, 17]). $\diamondsuit$

A *locally minimum-weight triangulation* is a triangulation such that for every edge shared by two triangles $\triangle(a,b,c)$ and $\triangle(a,b,d)$, edge bd is the shortest diagonal of the quadrilateral with vertices a, b, c, d. Locally minimum-weight triangulations have been extensively studied for their relationship to minimum-weight triangulations (see, e.g., [15]).

Theorem 17 *There exists an optimal N-checker for locally minimum-weight triangulations that runs in linear time and has degree* 2*.*

The optimal time complexity of the N-checkers of Theorems 15, 16, and 17 relies on the fact that $\lambda(G) = 2$ for the underlying graph of a triangulation. The next lemma, shows that for convex planar subdivisions, $\lambda(G)$ is simply exponential.

Lemma 13 *Let G be the underlying graph of a convex planar subdivision. Then the number $\lambda(G)$ of topologically distinct planar orderings of G is $O(2^n)$, where n is the number of vertices of G.*

Proof: The proof is based on the results of [7, 8], where the SPQR-trees of planar graphs that admit a planar straight-line drawing with convex faces are characterized. $\diamondsuit$

By combining Theorem 14, Lemma 12, Lemma 13, and Theorem 12 we obtain the following.

Theorem 18 *There exists an optimal N-checker for convex planar subdivisions that runs in linear time and has degree* 2*.*

Theorem 19 *There exists an N-checker for Delaunay diagrams that runs in linear time and has degree* 4*.*

6 Open Questions

Several questions remain open. Among them, we consider especially relevant the following.

1. Design effective T-checkers and N-checkers for other types of geometric graphs, such as Gabriel graphs, relative neighborhood graphs, and β-skeletons graphs.
2. Design a simple and practical T-checker for planar subdivisions that runs in linear time and has degree 2.
3. Design N-checkers for d-dimensional convex polytopes.

Acknowledgments

We would like to thank Giuseppe Di Battista, David Kirkpatrick, Michael Goodrich, and Luca Vismara for useful discussions.

References

[1] N. Amenta. Directory of computational geometry software. `http://www.geom.umn.edu/software/cglist/`.

[2] G. Barequet, S. S. Bridgeman, C. A. Duncan, M. T. Goodrich, and R. Tamassia. Classic computational geometry with GeomNet. In *Proc. Annu. ACM Sympos. Comput. Geom.*, 1997.

[3] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

[4] M. Blum and S. Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, Jan. 1995.

[5] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections.* Ph.D thesis, Universität des Saarlandes, Mar. 1996.

[6] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[7] G. Di Battista, R. Tamassia, and L. Vismara. On-line convex planarity testing. In *Proc. Graph-Theoretic Concepts in Computer Science*, Lecture Notes Comput. Sci. Springer-Verlag, 1994.

[8] G. Di Battista, R. Tamassia, and L. Vismara. On-line convex planarity testing. Technical Report CS-95-26, Dept. Computer Science, Brown Univ., 1995. `ftp://ftp.cs.brown.edu/pub/techreports/95/cs95-26.ps.Z`.

[9] G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM J. Discrete Math.*, 9(3):349–359, 1996.

[10] I. Fary. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged.*, 11:229–233, 1948.

[11] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. *Comput. Graph.*, 1996. Proc. SIGGRAPH '96.

[12] F. Harary. *Graph Theory.* Addison-Wesley, Reading, MA, 1972.

[13] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[14] D. G. Kirkpatrick. Establishing order in planar subdivisions. *Discrete Comput. Geom.*, 3:267–280, 1988.

[15] C. Levcopoulos and A. Lingas. On approximation behavior of the greedy triangulation for convex polygons. *Algorithmica*, 2:175–193, 1987.

[16] G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. *SIAM J. Comput.* to appear.

[17] G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. In *Proc. Annu. ACM Sympos. Comput. Geom.*, 1997.

[18] K. Mehlhorn and S. Näher. *Checking Geometric Structures*, Dec. 1996. Manual.

[19] K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 159–165, 1996.

[20] K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. Manuscript, 1997.

[21] G. F. Sullivan, D. S. Wilson, and G. M. Masson. Certification of computational results. *IEEE Trans. Comput.*, 44(7):833–847, 1995.